

1. WebDoor Creator

![Egalware](images/LogoEgalware.svg "WebDoorCreator")

Table of Content

- 1. WebDoor Creator
- 2. WebDoor Creator Project
 - 2.1. Terminology
- 3. Overall Functional Schema
 - 3.1. (A) Public WebServer
 - 3.2. (B.1) Internal WebDoor Server
 - 3.3. (B.2) Redis Server
 - 3.4. (B.3) Main DB
 - 3.5. (B.4) EgwDoor Processor
 - 3.6. (C) EgwProd (Multi-ax Machine)
- 4. Hw specifications
- 5. Expected Use cases
 - 5.1. User registration / login
 - 5.2. User Role assignments
 - 5.3. Projects and Orders
 - 5.4. Order creation
 - 5.5. Base Library
 - 5.6. Door creation
 - 5.7. Time/Cost calculation
- 6. Document Revision

2. WebDoor Creator Project

WebDoorCreator is the web incarnation and evolution of Egalware's DoorCreator© software.

In particular those are expected use cases

- multi user setup
- concurrent access
- 3D web based output of designed door
- DDF format production (for later automatic CNC generation at machine facility)

This document aims to represent detailed information about system actual internals and functional implementation and is passible of revision with the actual software production phases (requirements, general develop, alpha testing, beta testing, staging and go-live).

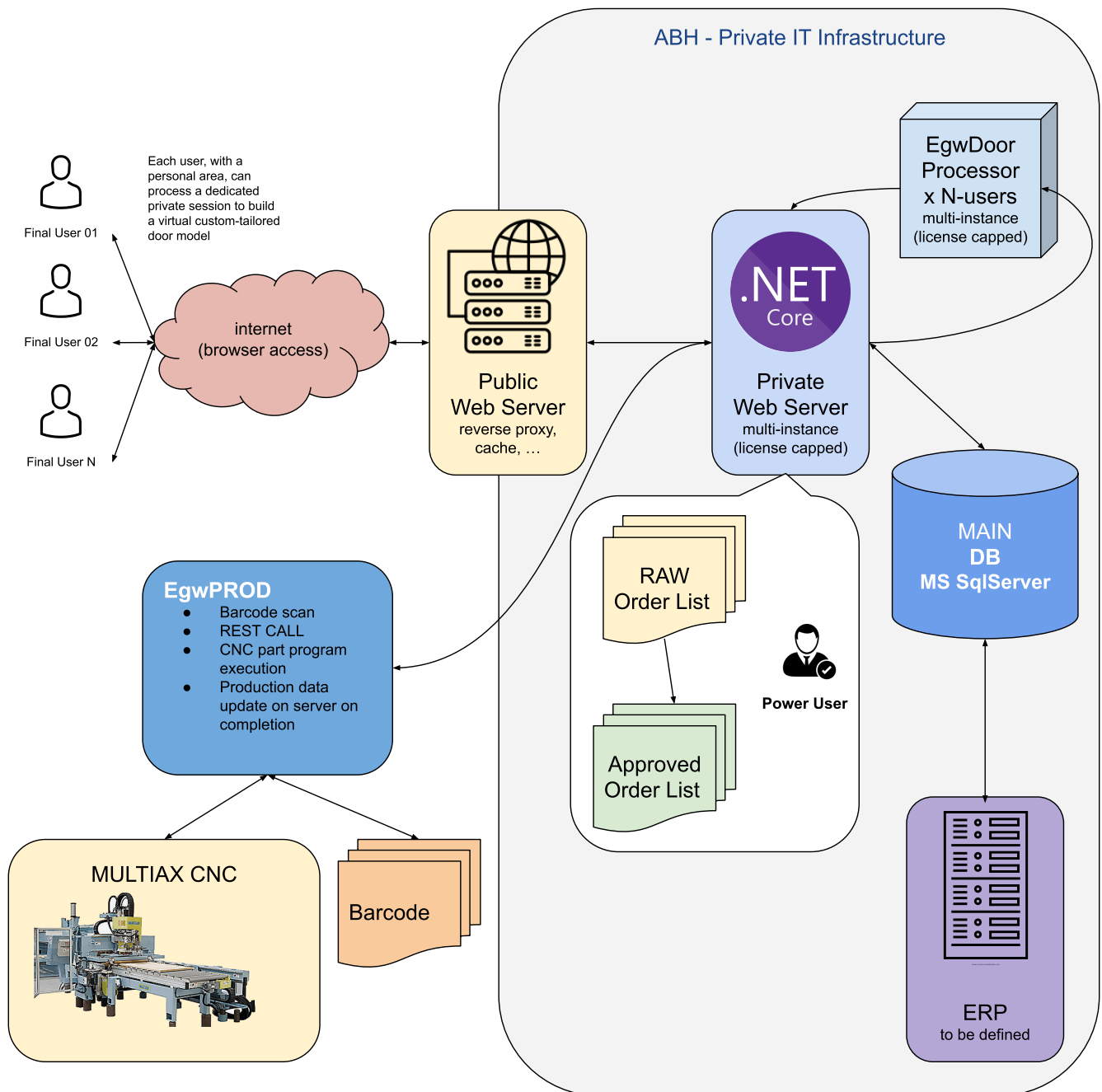
2.1. Terminology

Here a general introduction to terminology used for each component of the project, both for component and items to better understand subsequent details.

| Term | Description |
|--------------------------------|---|
| (A) Public WebServer | Reverse proxy, manages web access from final user decoupling access to local infrastructure |
| (B.1) Internal WebDoor Server | Actual main web server application (C# / .Net) |
| (B.2) Redis Server | High speed cache & messaging server for inter-application communication |
| (B.3) Main DB | Main Database for data storage, Microsoft SqlServer |
| (B.4) EgwDoor Processor | EgtCam5 based application for processing actual CAD/CAM/3D specific tasks |
| (C) EgwProd (Multi-ax Machine) | Main Workstation for specific door-data processing |
| WF_OrderMan | WorkFlow for actual order management: to be better defined |
| ERP | WorkFlow needs to be defined |
| EgwProd | Local Application on Multi-ax Machine for production execution |
| ProdCode | Generic unique code (with barcode/QRCode/DataMatrix) to be used from remote Machine to scan and execute automatic production |
| DDF | Door Definition Format , the base data-format to describe Doors (measures, attributes, hardware, ...) in order to generate CAD/CAM/3D/ISO program and each format requested for the solution |
| Project | Each Customer will be able to create project, consisting of 1+ doors defined in each project; each project will be managed as a whole (from approval to production/billing phase) |

3. Overall Functional Schema

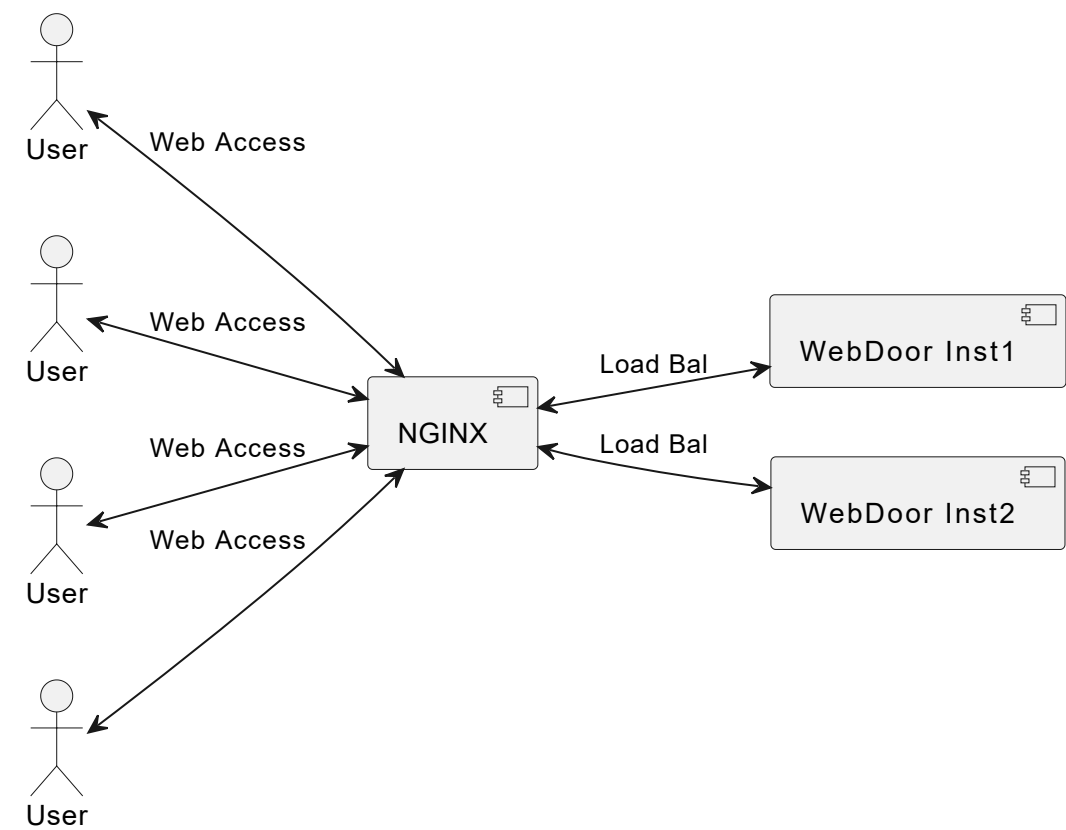
This schema is the general representation of the solution, better defined in subsequent chapters.



3.1. (A) Public WebServer

The public webserver will act as a reverse proxy (expected to use nginx) and load balancer for actual user access. It will decouple internal system to the external user web access optimizing static content caching & security. Setup of a standard single WAN Nginx linux virtual machine is included.

WebDoorCreator: Public WebServer



This way there will be secured access to internal IT infrastructure decoupled from external internet.

3.2. (B.1) Internal WebDoor Server

The internal WebDoor Server is a Blazor C# application, made to be hosted in one of this alternative scenario, to be discussed with ABH IT people:

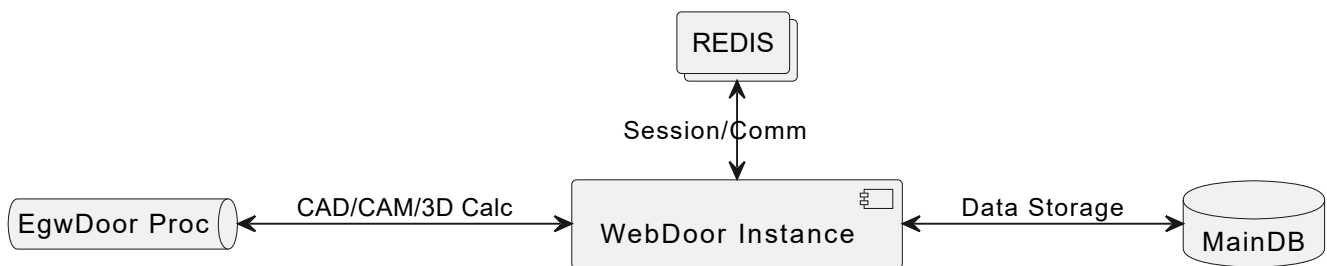
- IIS managed windows websites for c# .net (standard windows scenario)
- Linux virtual machine with dotNet framework kestrel server hosted as a services (normally the most performing scenario)
- Docker dotNet instances (at the moment is the deploy we used in less cases, but for this specific task might better fit)

We expect to deliver more than 1 instance of this server, both for redundancy and scalability (to give final user best performances we expect to have many instance, each managing many final-user session)

Final note: while the load balancer can fit also a mixed setup (IIS, Linux, docker), this is normally less user-friendly and slow in updating.

Each single instance will run as follow:

WebDoorCreator: Internal WebDoor Server



3.3. (B.2) Redis Server

The REDIS server will act as a cache / high speed data layer for example for managing multiple session with the cluster of WebDoor Instances and, at the same time, to act as a inter-process channel manager to permit data exchange between each solution component.

It is a RAM based server and must be kept secure in the "green zone" of the customer IT datacenter.

WebDoorCreator: Internal WebDoor Server



3.4. (B.3) Main DB

The Main DB server will be, as a first choice (and if not otherwise requested by customer) a Microsoft SqlServer Express instance. Can be swapped for a MariaDB (MySQL) solution, in particular in case of cluster DB requirements (high availability and if concurrent data access requirements exceed SqlServer capacity), but at the moment we expect MS SqlServer is the best fit.

3.5. (B.4) EgwDoor Processor

This component, based on **EgtCAM5** Kernel and libraries, is the main core of CAD/CAM/3D task execution.

This will be linked (with both REST webservice + REDIS channels/cache) to the (B.1) WebDoor Server.

In particular the WebDoor will send DDF online-generated file (derived from user-input via web-interface) to the EgwDoor Processor. This processor will generate in (near) realtime various output format upon request (base geometry, 3D web based artifacts, simulated ISO path for CAM calculation, ...).

The produced 3D (javascript based) web artifacts will be presented to the user in order to evaluate the final expected product during web-based door definition wizard.

3.6. (C) EgwProd (Multiax Machine)

For each Multiax machine put in place, there will be a local copy of the standard EgwProd software (enabled to read barcode/qr code/datamatrix data) and a copy of DoorCreator software (enabling realtime part program generation based on DDF final-product specification with respect on current tools and unique machine setup).

The software, after recognizing the unique code for each single door, will be able to call (B.1) WebDoor Server to receive DDF file corresponding to each unique door. A backup solution (eg taking into account the case of network connectivity problems/disconnection) can be put in place in the form of distributed/replicated "approved DDF file storage" in order to replicate on each production facility the set of near term part production (to be defined in detail, for example next 2-7 days of scheduled production).

4. Hw specifications

Hw requirements must be considered a work-in process based on general recommendation. Those are reserved for specific IT infrastructure for the current project and that can (and must) be setup in order to get best result

| Component | Virt | Num | Critical resource | Hw requirements |
|-------------------------------|------|-----|--------------------------|--|
| (A) Public WebServer | Y | 1+ | Bandwith, Latency | Linux Nginx, 2vCPU/2GB RAM each, 1Gbit Fiber + static IP |
| (B.1) Internal WebDoor Server | Y | 1+ | Processing (RAM/CPU) | 4+ vCore / 4GB RAM, Windows/Linux/Docker (to be tested) |
| (B.2) Redis Server | Y | 1 | RAM | Linux redis server, 4 vCpu, 4GB Ram |
| (B.3) Main DB | Y | 1 | Storage | Windows server + MsSqlServer, high speed storage (SSD min, better NVME), 4 vCPU (min), 8GB RAM (min) |
| (B.4) EgwDoor Processor | N | 1 | Processing (CPU/GPU/RAM) | Min 16Gb RAM, min 8 core CPU, high speed storage (SSD/NVME) + 8Gb videocard (min NVidia 3060 class expected) |

special notes:

- Virt: Y = can be a virtual server, N = phisical asset
- Num = number of item (1 single node, 1+ cluster for horizontal scaling)
- (B.3) DB: conditioned to final speed benchmarking an alternative option /MariaDB cluster) will be evaluated if bottleneck consideration might concern, HW will be subsequently adapted

5. Expected Use cases

Here's the brief description of use case and expected system behaviour.

5.1. User registration / login

User registration process will be based on email: each user must have a unique email (eg: used for password retrieval and user 2FA confirmation) and from email / tokenized-link exchange process the process will be carried out.

Each user must be registered (self registered or initially mass imported) before being given role on the platform.

5.2. User Role assignments

The platform will be managed through RBAC policy (Role-Based Access Control). So each user will be granted 1+ roles. Each role will permit single operation or multiple operation on the system.

The role levels will be (at minimum) the following:

| Role | Level | Parameter | Description |
|-----------------|-------|------------------------|--|
| User | 0 | - | Standar user, only access on platform |
| Customer | 1 | Customer Code | Group used to represent a final customer, representing the company = Door destination |
| CustomerUser | 2 | Customer Code + UserId | User associated with a Customer, can see/manage all customer orders |
| CustomerAdmin | 2 | Customer Code + UserId | SPECIAL User with highest privileges for Customer, can add users to group and approve orders |
| AbhOrderManager | 3 | UserId | SPECIAL User, from ABH, that is responsible for final customer's order acceptance |
| AbhAdmin | 4 | UserId | SPECIAL User, from ABH, that is responsible for ABH user management |
| SuperAdmin | 5 | UserId | General platform superuser for maintenance/setup task |

5.3. Projects and Orders

As already stated

Each Customer will be able to create project, consisting of 1+ doors defined in each project; each project will be managed as a whole (from approval to production/billing phase)

So the object hierarchy will be

- Order ACB.02
 - Door code 123 (x10 copy)
 - Door code 345 (x3 copy)
- Order DEF.01
 - Door code 263 (x6 copy)
 - Door code 785 (x5 copy)

Each item can be in several states, for example:

- Draft
- Requested (customer)
- Estimated price/time (ABH)
- Purchased (customer) | **frozen** / stop modification from customer
- Confirmed (ABH)
- Planned (ABH) | **frozen** / stop modification from ABH
- InProgress (ABH)
- Delivered / Billed / Closed

Normal workflow will be in forward direction only, returning to previous state will require escalation/approval (to be better defined).

5.4. Order creation

Each **Customer User** will be able to create a **project** or use an already available one in a "pre purchase" state.

When the order is completed, customer will send it to ABH for Estimation confirmation (because automatic costing can be done but final check for cost and expected delivery time is required to be user-driven).

After Estimation stage, purchase must be carried on from customer (with standard and already in-place company workflow of order management).

Resource allocation will be reported to ERP (materials + machine time, to be defined)

Finally resource planning and final estimation for delivery will be sent back to customers.

Production will be driven internally by ABH with support of the planned scheduling and actual machine task will be reported to the central system to record order fulfill and trigger logistics task on completion.

5.5. Base Library

Like for actual DoorCreator software, our automated part-program generation process requires definition of hardware in form of drawing and parameters in the form of a user-generated items library the system will consider in automatic program generation.

The process will be carried out with standard DoorCreator software instances already available to the customer.

5.6. Door creation

The process of defining the final expected door will be carried out with web-based interface.

In particular a wizard multi-step process, similar to the actual process implemented on "single-user" DoorCreator software.

The process will define at least these data:

- Door general dimension
- opening side (SX/DX)
- hardware
- items selection from **Base Library**
- ...

As a final product a DDF file will be real-time generated upon each data input, sent to the EgwDoor Processor for calculation and results will be presented to the user, in both form of 3D we-based content for display and data produced (feasibility, time/cost expectation, ...)

5.7. Time/Cost calculation

During standard user process the task of time/cost estimation (based on actual DDF model) will be carried out.

In particular we expect to have some data auto-calculated (ex time from CAM), while other data will be required to external system (ex: materials cost, availability date for materials or machine time, cost markup to define final user price, custom discount based on customer special agreements...)

6. Document Revision

| Date | Edit | Version | Note |
|------------|----------------|---------|------------------|
| 2022.11.03 | S.E. Locatelli | 0.1 | Initial draft |
| 2022.11.09 | S.E. Locatelli | 0.2 | Draft completion |
| 2023.03.21 | Z. Majid | 0.4 | Grammar Fixes |